



# Integration New Media's SecureNet Xtra

---

**For Macromedia Authorware**

User Manual Version 1.3

integration



new media © Integration New Media Inc. 2003 | Manual Version 1.3 | 2003-07-18

# Contents

---

<b>Contents</b>	<b>2</b>
<b>License agreement</b>	<b>4</b>
<b>Introduction</b>	<b>6</b>
Before you read this manual	6
Model of SecureNet Xtra	6
Easy to deploy applications	6
Tutorials and Samples	6
<b>Installation and startup</b>	<b>8</b>
System requirements	8
Limitations of SecureNet Xtra on Mac	8
Installation	8
How to register your SecureNet Xtra license	9
<b>SecureNet Xtra Basics</b>	<b>10</b>
What is a proxy server?	10
Proxy server setting	10
Defining Internet Explorer proxy server settings	11
Initializing the proxy server settings	11
Secure connection outside browser (HTTPS)	12
Basic network communication tasks	13
Formatting URLs	13
<b>Using SecureNet Xtra functions</b>	<b>14</b>
Requesting data	14
Sending data	15
Monitoring the progress of requests	15
Handling Errors	16
<b>Troubleshooting network communications</b>	<b>18</b>
Troubleshooting proxy server connections	18
<b>Delivering to the end user</b>	<b>20</b>
Playback from a packaged piece	20
On Windows	20
On Mac OS X	20
On Mac OS 9	20
Playback from a web-packaged piece	20
<b>Where to go next</b>	<b>23</b>
Tech Notes	23
Samples and tutorials	23
Latest user manual	23



Discussion list	23
Technical support	23
<b>Appendix 1: Functions reference</b>	<b>24</b>
snxDownloadNetThing	24
snxGetNetText	25
snxGetStreamStatus	26
snxNetAbort	27
snxNetDone	27
snxNetError	27
snxNetTextResult	28
snxPostNetText	28
snxProxyServer	29
<b>Appendix 2: Error codes</b>	<b>31</b>
<b>Index</b>	<b>39</b>

Macromedia, Authorware, Director and Xtra are trademarks or registered trademarks of Macromedia, Inc. in the United States and/or other countries. Microsoft, Internet Explorer, Windows and Windows NT are trademarks or registered trademarks of Microsoft Corporation, registered in the U.S. and/or other countries.

Other trademarks, trade names and product names contained in this manual may be the trademarks or registered trademarks of their respective owners, and are hereby acknowledged.



# License agreement

---

PLEASE READ THIS LICENSE AGREEMENT CAREFULLY BEFORE USING Integration New Media's SecureNet XTRA. BY USING Integration New Media's SecureNet XTRA, YOU AGREE TO BECOME BOUND BY THE TERMS OF THIS LICENSE AGREEMENT.

The enclosed computer program(s), license file and data (collectively, "Software") are licensed, not sold, to you by Integration New Media, Inc. ("INM") for the purpose of using it for the development of your own products ("Products") only under the terms of this Agreement. INM and its licensors reserve any rights not expressly granted to you. INM and its licensors grant you no right, title or interest in or to the Software. The Software is owned by INM and its licensors and is protected by International copyright laws and international treaties.

## 1. License.

- (a) You may install one copy of the Software on a single Windows-compatible computer and one copy on a single Macintosh-compatible computer. To "install" the Software means that the Software is either loaded or installed on the permanent memory of a computer (i.e., hard disk). This installed copy of the Software may be accessible by multiple computers, however, the Software cannot be installed on more than one computer at any time. You may only install the Software on another computer if you first remove the Software from the computer on which it was previously installed. You may not sublease, rent, loan or lease the Software.
- (b) You may make one copy of the Software in machine-readable form solely for backup purposes. As an express condition of this Agreement, you must reproduce on each copy any copyright notice or other proprietary notice that is on the original copy supplied by INM.
- (c) Your license is limited to the particular version (collectively "Version") of the Software you have purchased. Therefore, use of a Version other than the one encompassed by this License Agreement requires a separate license.
- (d) Upon purchase of a Software license, you are given a license number, which is subject to the restrictions set forth above and may not be distributed by you in any way. However, INM and its licensors grant you a royalty-free right to reproduce and distribute the file named "SecureNetXtra.x32" ("Runtime Kit") provided that (i) you distribute the Runtime Kit only in conjunction with and as part of your own Products; (ii) own a license for the specific Version of the Software that contains the Runtime Kit; (iii) agree to indemnify, hold harmless and defend INM and its licensors from and against any claims or lawsuits, including attorney's fees, that arise or result from the use or distribution of your Products with the Runtime Kit.
- (e) Any third party who will use the Runtime Kit in an authoring environment must purchase its own license of the Software.
- (f) Notwithstanding any other terms in this License, if the Software is licensed as an upgrade or update, then you may only use the Software to replace previously validly licensed versions of the same software. You agree that the upgrade or update does not constitute the granting of a second license to the Software (i.e., you may not use the upgrade or update in addition to the software it is replacing, nor may you transfer the software which is being replaced to a third party).

## 2. Restrictions.

- (a) The Software contains trade secrets in its human perceivable form and, to protect them, you may not MODIFY, TRANSLATE, REVERSE ENGINEER, REVERSE ASSEMBLE, DECOMPILE, DISASSEMBLE OR OTHERWISE REDUCE THE SOFTWARE TO ANY HUMAN PERCEIVABLE FORM. YOU MAY NOT MODIFY, ADAPT, TRANSLATE, RENT, LEASE, LOAN OR CREATE DERIVATIVE WORKS BASED UPON THE SOFTWARE OR ANY PART THEREOF.
- (b) THE SOFTWARE IS NOT INTENDED FOR USE IN THE OPERATION OF NUCLEAR FACILITIES, AIRCRAFT NAVIGATION, AIR TRAFFIC CONTROL, OR OTHER ENVIRONMENTS IN WHICH THE FAILURE OF THE SOFTWARE COULD LEAD TO DEATH, PERSONAL INJURY, OR SEVERE PHYSICAL OR ENVIRONMENTAL DAMAGE.
- (c) You may not transfer or assign your rights under this License to another party without INM's prior written consent. Assignment application forms can be obtained from INM's sales department.

**3. Copyright notices.** You may not alter or change INM's and its licensors' copyright notices as contained in the Software. You must include a copyright notice, in direct proximity to your own copyright notice, in substantially the following form "Portions of code are Copyright ©2002 used under license by Integration New Media, Inc."

**4. Acceptance.** The Software shall be deemed accepted by you upon delivery unless you provide INM, within two (2) weeks therein, with a written description of any bona fide defects in material or workmanship.

**5. Termination.** This Agreement is effective until terminated. This Agreement will terminate immediately without notice from INM or judicial resolution if you fail to comply with any provision of this Agreement. Upon such termination you must destroy the Software, all accompanying written materials and all copies thereof, and Sections 6 and 7 will survive any termination.



6. **Limited Warranty.** INM warrants for a period of ninety (90) days from your date of purchase (as evidenced by a copy of your receipt) that the media on which the Software is recorded will be free from defects in materials and workmanship under normal use and the Software will perform substantially in accordance with the manual. INM's entire liability and your sole and exclusive remedy for any breach of the foregoing limited warranty will be, at INM's option, replacement of the disk, refund of the purchase price or repair or replacement of the Software.

7. **Limitation of Remedies and Damages.** In no event will INM, its parent or subsidiaries or any of the licensors, directors, officers, employees or affiliates of any of the foregoing be liable to you for any consequential, incidental, indirect or special damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of business information and the like), whether foreseeable or not, arising out of the use of or inability to use the Software or accompanying written materials, regardless of the basis of the claim and even if INM or an INM representative has been advised of the possibility of such damage. INM's liability to you for direct damages for any cause whatsoever, and regardless of the form of the action, will be limited to the greater of US \$199.00 or the money paid for the Software that caused the damages.

THIS LIMITATION WILL NOT APPLY IN CASE OF PERSONAL INJURY ONLY WHERE AND TO THE EXTENT THAT APPLICABLE LAW REQUIRES SUCH LIABILITY. BECAUSE SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

8. **General.** This Agreement will be construed under the laws of the Province of Quebec, except for that body of law dealing with conflicts of law. If any provision of this Agreement shall be held by a court of competent jurisdiction to be contrary to law, that provision will be enforced to the maximum extent permissible, and the remaining provisions of this Agreement will remain in full force and effect.

9. The parties acknowledge having requested and being satisfied that this Agreement and its accessories be drawn in English. Les parties reconnaissent avoir demandé que cette entente et ses documents connexes soient rédigés en anglais et s'en déclarent satisfaits.



# Introduction

---

Integration New Media's SecureNet Xtra enables you to create Authorware pieces that can:

- Access Internet/Intranet resources even from behind proxy servers that may or may not require username and password authentication;
- Access Internet/Intranet resources from secure web sites using the HTTPS protocol from both Authorware's runtime environment and the Web Player.

In addition to these unique features, SecureNet Xtra helps you reduce costs by allowing you to debug HTTPS queries right from Authorware's authoring environment.

## Before you read this manual

This user manual provides the essential information you need to use SecureNet Xtra. The section entitled SecureNet Xtra Basics outlines the typical scenarios in which SecureNet Xtra is used.

## Model of SecureNet Xtra

SecureNet Xtra was built on the model of Director's NetLingo. SecureNet Xtra provides nearly all the functions found in NetLingo and the function names and syntaxes are almost identical. The only visible difference is the "snx" at the beginning of all SecureNet Xtra function names. Therefore, if you are familiar with Director's NetLingo, you will find SecureNet Xtra easy to use.

## Easy to deploy applications

Your packaged Authorware pieces can be delivered on CD-ROM, or downloaded onto the end-user's hard drive from the Internet. The end-user doesn't need any special player.

You don't even have to know your user's proxy server settings. At authoring time you can choose to allow SecureNet Xtra to detect the IP address and port number for a proxy server based on the user's current Internet Explorer settings. This also means the user's IT staff doesn't have to become involved in deploying your solutions. See System requirements for complete specifications.

## Tutorials and Samples

In addition to the Xtra itself, INM provides a set of samples, tutorials and documentation to assist you.

The **First Steps** tutorial gives you step-by-step instructions on how to use SecureNet Xtra to complete a sample application that demonstrates how SecureNet Xtra can be used in an e-learning setting, to send and retrieve test scores over a secure connection.

With the **Proxy-Connect** tutorial you learn how to use SecureNet Xtra to connect from a computer behind a proxy server, to a World Wide Web server that returns data.



Both of these tutorials come with sample Authorware pieces that you can use and modify for your own projects.

Download the First Steps and Proxy-Connect tutorials from our web site:

<http://www.IntegrationNewMedia.com/products/snxauthorware/demos/>



# Installation and startup

---

## System requirements

- Windows: Windows 95/98/ME/NT4/2000/XP.
- Macromedia Authorware version 4.x, 5.x, 6.x, 7.x
- Microsoft Internet Explorer 4.x and later (or at least the corresponding wininet.dll library).
- Authoring on Mac Classic OS 9.1 and higher, via Authorware 4.
- Playback on Mac Classic OS 9.1 and higher, via Authorware Packager version 4 - 6.5 and AW Player version 4 - 6.5. Requires URL Access Manager library v2.2.1 or higher (URL Access Manager v2.4.1 is recommended).
- Playback on Mac OS X, version 10.2.5 and higher, via Authorware 7 Packager and AW 7 Player

For details on how to publish for Mac playback, see Delivering to the end user.

## Limitations of SecureNet Xtra on Mac

### HTTPS across a proxy server with authentication

- On Mac OS 9.x, when performing a secure HTTP operation (<https://>) across a proxy that requires authentication, `snxGetStreamStatus()` cannot be used to display the status of the request.
- On Mac OS X, secure HTTP communications bypass the proxy server. When a direct Internet connection is present the request goes directly through to the URL being queried.

## Installation

- 1 Make sure Authorware is not running.
- 2 Place the Xtra (a file named *“SecureNetXtra.x32”*) into the Xtras folder. This folder is located in the same place as the Authorware application.
- 3 Launch Authorware.
- 4 You can learn which Xtras are available to Authorware by opening the Functions window from the Window menu and checking the content of the Category menu. The available Xtras are listed right after the “Xtra’s(All)” item.  
If SecureNet Xtra is installed, you should see SecureNet Xtra listed in the Category menu as well as all other available Scripting Xtras. Note that this technique applies to Scripting Xtras only.





## How to register your SecureNet Xtra license

**Note:** You must call the `snxRegister()` function in the beginning of each Authorware piece that uses SecureNet Xtra.

The free evaluation version of SecureNet Xtra contains all the functionality of the licensed version, except that a Demo Version alert appears the first time the piece calls a SecureNet Xtra function. In order to ship a product containing SecureNet Xtra you must purchase a license for the software.

Please make sure to read and agree to the License Agreement (see License agreement) before purchasing your SecureNet Xtra license. To purchase a license, visit our online store at:

<http://www.IntegrationNewMedia.com/store/>,

or contact us by phone at:

+1 800 400 1772 or +1 514 871 1333, Option 5.

**Note:** Never distribute the license number. If your piece must be sent to other developers, first remove the license number from your script.

To register your Xtra, and remove the Demo splash screen, you need to call the SecureNet Xtra function `snxRegister(thekey)` at the beginning of each Authorware that uses the Xtra. The parameter *theKey* is the license number that was given to you by INM, upon purchase of the license, surrounded by quotes. To avoid making mistakes we recommend you copy/paste this number from the email that Integration New Media sends you.

Example: `snxRegister("S1R9-T6K8-2HD2-J391")`



## SecureNet Xtra Basics

---

You would use SecureNet Xtra in these scenarios:

- 1) Your application crosses a proxy server that requires authentication
- 2) Your application requires a secure connection (HTTPS) that runs from a within or outside the browser.
- 3) Your application meets both of these requirements

### What is a proxy server?

A proxy server intercepts requests between a client (e.g., your desktop) and a web server. Many corporate offices now use secure proxy servers for one or more of the following reasons:

- to mask the identity of the client computer; only the proxy server's IP address is seen from outside.
- to serve web pages from its cache and thus alleviate network traffic when the same URLs are requested repeatedly
- to control access to the web

Depending on a proxy server's security settings, it may require a username and password; this is known as authentication. As a multimedia developer, you must be aware that more and more potential clients are located in environments where they are behind proxy servers that may require username and password authentication before allowing any http communication. Therefore, if your application requires Internet access, you have to address this issue.

See the following definitions of "proxy server" for further clarification:

[http://www.newworldpartnership.com/def\\_proxy.html](http://www.newworldpartnership.com/def_proxy.html)

[http://www.webopedia.com/TERM/p/proxy\\_server.html](http://www.webopedia.com/TERM/p/proxy_server.html)

### Proxy server setting

In this setting, the end-user is typically behind a proxy server that requires username and password authentication before allowing any http communication.

Many corporate offices now use secure proxy servers. Corporate e-learning applications often need to be used by company members from both behind the proxy server (at work), and externally (on the road). If an e-learning application obtains content from outside the proxy, the internal users need SecureNet Xtra to be able to receive it. Likewise if the application sends tracking information to a database that is behind the proxy, SecureNet Xtra must be used to allow data to pass into the secure server environment from wherever the application is run.

Depending on the proxy server's security settings, it may require a username and password to be sent before making any external communication. SecureNet Xtra supports Microsoft Internet Explorer's encryption schemes to send an encoded username and password that the proxy server is able to



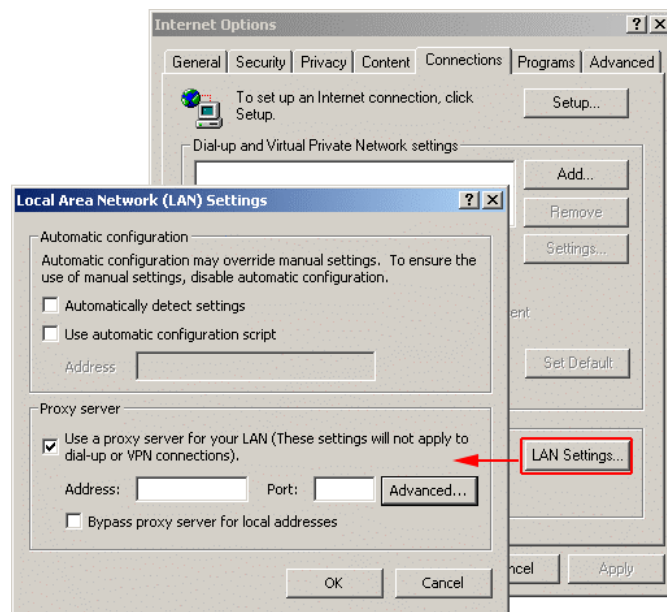
decode. It can also use the proxy Address and Port settings defined in Internet Explorer on Windows.

## Defining Internet Explorer proxy server settings

To set or check the proxy server settings used by Internet Explorer:

- 1 Click **Tools > Internet Options...** from the Internet Explorer menu bar.
- 2 In the **Connections** tab of the **Internet Options** dialog box, click **LAN Settings...**
- 3 The lower half of the LAN settings window contains **Proxy server** settings.
- 4 If you connect to the Internet using a proxy server, the checkbox should be checked and the **Address** and **Port** fields should be filled in with the correct settings.
- 5 Click **Advanced...** to view or set specifications for multiple proxy servers.

IE Proxy server settings



## Initializing the proxy server settings

You need to call the SecureNet Xtra function `snxProxyServer()` before using any of the other SecureNet Xtra functions. This function lets SecureNet Xtra know that you will be using a proxy server, and identifies the proxy server *IP Address* and *Port number*, and the *Username* and *Password*, if required.

Examples:

```
cal l Object ("snxProxyServer", "204.19.171.45", 8080, "Vahe",  
"secretpwd")  
cal l Object ("snxProxyServer", Default, "Vahe", "secretpwd")
```



Once `snxProxyServer()` is called to set the proxy server parameters, all subsequent SecureNet Xtra functions use these proxy server settings. When you want to stop using the proxy server, you can call `snxProxyServer` with the `#Stop` parameter, as in:

```
call Object ("snxProxyServer", #Stop)
```

The `snxProxyServer` function can also be used to obtain information about the proxy server settings in use. For more details see `snxProxyServer` in Appendix 1: Functions reference.

## Secure connection outside browser (HTTPS)

For this scenario, your end-user is not necessarily behind a proxy, but your application requires a secure connection, with encrypted data transfer.

An example of this application is a course delivered online or on CD-ROM where the students' ids and scores are confidential and that data must be sent to the course administrator or a database on a secure web server.



## Basic network communication tasks

The basic tasks that SecureNet Xtra allows you to do are:

- **Request** data (either text or encrypted files from a URL)
- **Send** data (either text or encrypted files to a URL)
- **Monitor** the status of the connection to a remote URL

## Formatting URLs

When you use SecureNet Xtra functions to request or send data over the Internet, the URLs being accessed must be properly formatted, according to the following rules:

- They must start with “http://” or “https://”.
- If username and password are required by the destination server, they must be included in the URL using this syntax:  
“https://myUsername: myPassword@www. SecureBuy. com/Store. html ”
- URLs must be encoded. They cannot contain non-ASCII characters, ASCII control characters, reserved characters, or unsafe characters. The following table lists the characters that must be encoded for SecureNet Xtra URLs, with their hexadecimal representations. To encode a character within a URL, replace it by the “%” symbol, followed by the two-digit hexadecimal representation.

Character	Hexadecimal representation
<b>Reserved characters</b>	
Forward slash/Virgule (“/”)	%2F
Colon (“:”)	%3A
'At' symbol (“@”)	%40
<b>Unsafe characters</b>	
Space	%20
Quotation marks	%22
Percent (“%”)	%25
Caret (“^”)	%5E
Tilde (“~”)	%7E
Left square bracket (“[”)	%5B
Right square bracket (“]”)	%5D
Grave accent (“`”)	%60



## Using SecureNet Xtra functions

---

### Requesting data

You initiate a request for data from a remote server using either `snxGetNetText()` or `snxDownloadNetThi ng()`. `SnxGetNetText()` allows you to specify the URL for a text file, ASP or CGI script or web page and obtains the result as text. `SnxDownloadNetThi ng()` lets you download a file that is either text or non-text, and save it to a specified location.

After initiating a request, your application calls `snxNetDone()`, with the ID of the request, to determine if the result has been returned.

When `snxNetDone` returns a value of `TRUE`, you must call `snxNetTextResul t()` to obtain the text data returned by the remote server after initiating a request with `snxGetNetText()`. You can then display the result or use it in your code.

#### Example:

```
NetID := snxGetNetText("http://www.SecureNetXtra.com")
-- check for errors in initiating the request
if NetID < 0 then
    theError := NetID
    GoTo(IconID@"GetError Message")
end if

-----

-- loop until the request has completed
repeat while (snxNetDone(NetID) = False)
end repeat

theError := snxNetError(NetID)
--if error occurs, display the error and abort connection
if theError <> "OK" then
    GoTo(IconID@"GetError Message")
end if

-----

RetString := snxNetTextResul t(NetID)
GoTo(IconID@"Di spl ayResul t")
```

When using `snxDownloadNetThi ng()`, you do not have to call `snxNetTextResul t(NetID)`. As soon as `snxNetDone(NetID)` is `TRUE`, the file should be downloaded to the local machine. Note that `snxDownloadNetThi ng()` cannot be used in a web-packaged piece within a browser; SecureNet Xtra will return error code -2101.



## Sending data

`snxPostNetText()` allows you to send or “post” data to a HTTP or HTTPS server. For example, you would use `snxPostNetText()` to submit form data. As with `snxGetNetText()`, your application calls `snxNetDone()`, with the ID of the post function, to determine if a result has been returned from the remote URL.

When `snxNetDone` returns a value of `TRUE`, you must call `snxNetTextResult()` to obtain the text data returned by the remote server. You can then display the result or use it in your code.

### Example:

```
NetID := snxPostNetText (http://www.
    integrationnewmedia.com/orderform.cgi, [#color: "red",
    #quantity: "200"])
-- check for errors in initiating the post
if NetID < 0 then
    theError := NetID
    GoTo(IconID@"GetError Message")
end if

-----

-- loop until the request has completed
repeat while (snxNetDone(NetID) = False)
end repeat

theError := snxNetError(NetID)
--if error occurs, display the error and abort connection
if theError <> "OK" then
    GoTo(IconID@"GetError Message")
end if

-----

-- optionally display the returned message
RetString := snxNetTextResult(NetID)
GoTo(IconID@"DisplayResult")
```

## Monitoring the progress of requests

After initiating a request or sending data to a remote URL, you need to check to see that the request was received before you can display or do anything with result returned.

To find out when a network request has completed, call `snxNetDone(NetID)`. It returns `True` if the request has completed (with or without errors), or `False` if the operation is still in progress.

`SnxGetStreamStatus(NetID)` allows you to monitor the status of any SecureNet Xtra network operation. It returns a `propertyList` containing the status of the network operation identified by `NetID`. The format of the property list is (`URL:"xxx", #state: "yyyy", #bytesSoFar: z, #bytesTotal: u, #error: e`), where

*#URL*: the current URL



*#state*: current state. The value can be Connecting, Started, InProgress, Complete or Error.

*#bytesSoFar*: number of bytes transferred.

*#bytesTotal*: total number of bytes to transfer. This value may be equal to *#bytesSoFar* if the server doesn't include the content length in the header.

*#Error*: Empty string if the download is not complete, "OK" if it completed successfully, or an error code (<0) if the operation ended with an error.

You can use the properties *#bytesSoFar* and *#bytesTotal* to display a progress bar to the user. See *snxGetStreamStatus* in Appendix 1, for a detailed example of how to show the status of a network operation.

## Handling Errors

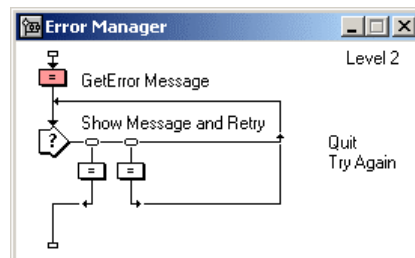
The SecureNet Xtra function *snxNetError*(NetID) simply returns an error code if an error occurred during the request specified by NetID. *Appendix 2: Error codes* contains a listing of each error code and its corresponding short and detailed descriptions.

You can script your own error handling in Authorware, or you can take advantage of the sample called "How to handle errors in SecureNet Xtra", in the Tech Notes section of INM's web site:

<http://www.IntegrationNewMedia.com/support/snxauthorware/technotes/>

In this sample, the Error Manager icon contains a script that translates the error code returned by SecureNet Xtra and displays the corresponding error description. Before entering the GetError Message calculation icon, the error code must be stored in a variable named **theError**. Your piece should only enter this icon if **theError** is not "OK".

Error Manager icon, Level 2,  
from the ErrorManager  
sample piece.



You may copy the entire icon into your own Authorware piece, and transfer control to it whenever the error code returned from a SecureNet Xtra operation is not "OK". Alternatively, you can just copy the code inside the "GetError Message" icon and create your own display icons.

### Example:

```
-- Loop until the request has completed
repeat while (snxNetDone(NetID) = False)
end repeat

--If an error occurs, display the error and retry connection
theError := snxNetError(NetID)
if theError <> "OK" then
    GoTo(IconID@"Error Manager")
end if
```





The two buttons labeled **Quit** and **Try Again** are provided to transfer control to different parts of the Authorware piece. You can insert statements to go to specific icons. For an example of how the Error Manager is used, download the FirstSteps tutorial from our website at:

<http://www.IntegrationNewMedia.com/products/snxauthorware/demos>.



# Troubleshooting network communications

---

Network communications may fail due to any of several possible conditions:

- internal communication problems in the user's network
- syntax error in the URL or in the parameters sent via a post
- invalid local file name for snxDownloadNetThing
- server dead
- proxy server not working or wrong username and password
- wrong or missing username and password in the URL for an https request

This section outlines how to test where the problem lies.

## Troubleshooting proxy server connections

The steps in this flowchart should be performed from within the proxy server network, usually on the end-user's computer. Before you begin, make sure you have the correct Proxy Address and Port number setting in Internet Explorer. See Defining Internet Explorer proxy server settings.

**Step 1.** INM provides a HTML file named testproxy.htm with the SecureNet Xtra package. Copy this file to the server your application connects to. Then type the URL of the HTML file (the one on your server) directly in Internet Explorer. Example:

`http://myserver/mytest/testproxy.htm`

### Does it work?

**Yes:** If the URL returns a message saying the “connection is working” it means there is no error in getting past the proxy server and the error must lie on the Authorware side, as opposed to within the network. Continue to Step 2.

**No:** If the result of Step 1 was not successful – you did not receive any feedback-- it means the problem lies within the internal network configuration or the proxy settings in Internet Explorer. Contact the System Administrator at the user's site.

**Step 2.** Play the Proxy-Connect sample piece available from:

<http://www.IntegrationNewMedia.com/products/snxauthorware/demos/>.

Replace the URL with the URL you used in Step 1.

### Does it work?

**Yes:** If this sample works, it means the problem lies within your Authorware scripts or server-side scripts. Continue to Step 3.

**No:** If this sample does not work, there may be an error in SecureNet Xtra. Contact INM's support team at:

<http://www.IntegrationNewMedia.com/support/>

**Step 3.** Try debugging your code. Check that the URL you are trying to access is properly encoded. See Formatting URLs.



If you need assistance, Integration New Media's Developer Assistance program can give you detailed support on your projects. Visit our Developer Assistance page on our web site to find out more about this service.

<http://www.IntegrationNewMedia.com/services/developer-assistance/>



## Delivering to the end user

---

SecureNet Xtra is designed in a way that minimizes any last minute changes needed before delivery to the end-user.

### Playback from a packaged piece

#### On Windows

If you are packaging an Authorware piece to be played on Windows, you can use the one-button publishing option in Authorware to create the package. This should copy the Xtra, named **SecureNetXtra.X32** to a folder named **Xtras** located in the same folder as your packaged piece.

#### On Mac OS X

If you are packaging an Authorware piece to be played on Mac OS X, you need to use the Authorware 7 packager and the end-user needs to have the Authorware 7 for Mac Player. The Mac OS X version of SecureNetXtra.XTR must be in a folder named Xtras located in the same folder as your packaged piece.

#### On Mac OS 9

If you are packaging an Authorware piece to be played on Mac Classic OS 9.1 or higher, you need to use the Authorware 6.5 (or lower) packager and the end-user needs to have an Authorware Player for Mac Classic. The Mac Classic OS version of SecureNetXtra.XTR must be in a folder named Xtras located in the same folder as your packaged piece.

If you are distributing your packaged piece on Mac OS 9.x, you must include the Mac binary Xtra, named **SecureNetXtra.XTR.bin** in this Xtras folder.

If you are distributing your packaged piece on Mac OS X, you must include the Mac binary Xtra, named **SecureNetXtra.OSX.bin** in this Xtras folder. This file comes with the SecureNet Xtra package.

### Playback from a web-packaged piece

SecureNet Xtra can be used in a web-packaged piece whether played locally or through the web. If you want your web piece to be accessible from both Mac OS X and Mac OS 9, you need to create two different web packages:

- Use the AW 7 Web Packager to create a web-packaged piece that can be played on Windows and Mac OS X (with the AW7 Player for Mac)
- Use the AW 6.5 (or lower) Web Packager to create a web-packaged piece that can be played on Windows and Mac OS 9.x, with the AW 6.5 or lower Player for Mac.

Note: The Mac OS X version of the Xtra and the Mac OS 9 version are both named **SecureNetXtra.XTR**. Therefore, take precautions to keep them separate and not mix them up. Check **Get Info** if you are not sure which file is which.



The Authorware Web Packager creates a map file (\*.aam), which instructs the Web Player to download the necessary Xtras from the web server to the end-user's machine. The Xtras must be available on the server, usually in a folder named XTRAS.

To ensure that SecureNet Xtra is accessible to a web-packaged piece on both Mac and Windows machines, make sure the map file contains the following lines:

For a Windows and Mac OS 9 web package, the "bin" line looks like this:

```
put XTRAS
bin win32 "SecureNetXtra.x32" "SecureNetXtra.X32"
bin mac "SecureNetXtra.xtr.bin" "SecureNetXtra.xtr" recycle,macbinary
```

For a Mac OS X web package, the "bin" line looks like this:

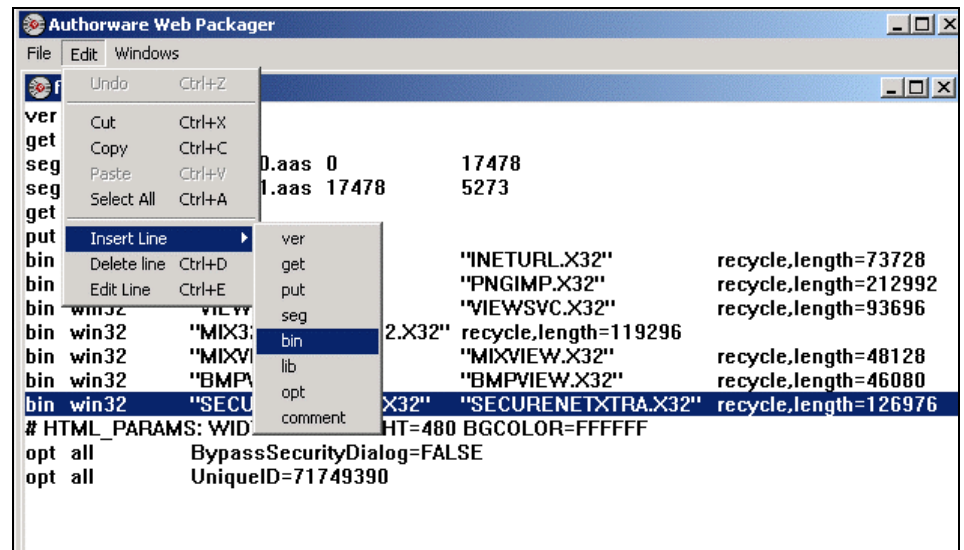
```
bin mac "SecureNetXtra.OSX.bin" "SecureNetXtra.xtr" recycle,macbinary
```

To ensure that the Xtra is downloaded on the Mac, you will have to edit the map file (.aam) and add the line for the Mac binary. You may cut and paste the lines above into the map file using any text editor.

Alternatively, you may edit the map file using the Web Packager as follows:

- 1 Double-click the .aam file to open the Authorware Web Packager, or open it from Window's **Start** menu.
- 2 You may have to click **File > Open** and browse to locate the .aam file.
- 3 To insert the line for the Mac binary, choose **Edit > Insert Line > bin**

Authorware 6 Web  
Packager: Insert line for  
Mac binary



- 4 In the Edit dialog box, choose Platform : **Mac**; Server:  
**SecureNetXtra.xtr.bin** (or **SecureNetXtra.OSX.bin** for Mac OS X), Local:  
**SecureNetXtra.xtr**
- 5 Check **MacBinary** and **Recycle**



Authorware 6 Web Packager  
Edit dialog options

Line type: bin OK Cancel

Platform: mac

File Names

Server: securenetextra.xtr.bin

Local: securenetextra.xtr

Flags

Preempt: ☐ OnDemand: ☐

Recycle: ☒ MacBinary: ☒

Optional Attributes

File Type: File Creator:

File Length:

You also have to manually place the Mac binary version(s) of the Xtra in the Xtras folder, within the Web folder of the web-packaged piece. These files, named **SecureNetXtra.xtr.bin** and **SecureNetXtra.OSX.bin**, are included with the SecureNet Xtra package. The Mac OS X and Mac OS 9 binary files are named differently so that there is no confusion on the web server. However, once downloaded and decompressed, they both contain files named SecureNetXtra.Xtr



## Where to go next

---

### Tech Notes

If you experience a problem with SecureNet Xtra, before you call Technical Support, please check the Tech Notes section on our Web site: (<http://www.IntegrationNewMedia.com/support/snxauthorware/technotes/>) to see if your problem is discussed there, and visit the Download section (<http://www.IntegrationNewMedia.com/download/>), to make sure you are using the most up-to-date version of SecureNet Xtra.

### Samples and tutorials

INM provides sample Authorware pieces and tutorials to help you get started quickly with SecureNet Xtra. Check the demos page on our web site at <http://www.IntegrationNewMedia.com/products/snxauthorware/demos/>.

### Latest user manual

Also, remember to read any relevant sections of the SecureNet Xtra User Manual. The user manual may be updated between releases of the product, to clarify or correct any misleading information. The latest version of the manual is available for download from INM's Download Library (<http://www.IntegrationNewMedia.com/support/snxauthorware/manuals/>).

### Discussion list

In any event, we recommend that you subscribe to SecureNetXtra-L, a discussion list dedicated to developers using SecureNet Xtra (check <http://www.IntegrationNewMedia.com/support/list/> for details).

### Technical support

INM offers free basic technical support to license owners as well as clients who are in the process of evaluating our products. However, **priority is given to license owners**. Make sure to provide your license number when you contact us to maintain your priority status.

To obtain support please fill out the support form on our web site:

<http://www.IntegrationNewMedia.com/support/support-form.asp>



## Appendix 1: Functions reference

---

### snxDownloadNetThing

Syntax	<code>snxDownloadNetThing (URL, LocalFile)</code>
Parameters	<i>URL</i> is the URL of the file to download. <i>localFile</i> is the full name (including the path) of the file on the local disk.
Return	Success: unique id for the operation (> 0) Failure: error code (<0)
Description	<p>Initiates the loading of a file on an HTTP or HTTPS server.</p> <p>The URL must be properly encoded and must start with “http://” or “https://”.</p> <p>To access a file on a HTTPS server, depending on how the server is configured, you may need to specify a user name and password within the URL, using the standard URL syntax.</p> <p>Use <code>snxNetDone</code> to find out when the <code>snxDownloadNetThing</code> operation is complete. Use <code>snxNetError</code> (or the Error Manager script) to find out if the operation was successful.</p> <p>Use <code>snxGetStreamStatus</code> to get the bytes downloaded so far.</p> <p><code>snxDownloadNetThing</code> is not supported in a browser (web-packaged pieces or Shockwave movies).</p>

#### Examples

```
-- simple file download from a remote URL
NetID := snxDownloadNetThing(
    "http://www.integrationsnewmedia.com/webmedia/logo.gif" ,
    FileLocation ^ "logo.gif")
repeat while (snxNetDone(NetID) = False)
end repeat
LogoFileName := FileLocation ^ "logo.gif")
-----
-- with username and password (accessing https server)
NetID := snxDownloadNetThing(
    "https://myUsername:myPassword@www.integrationsnewmedia.com/web
    media/logo.gif", FileLocation ^ "logo.gif")
repeat while (snxNetDone(NetID) = False)
end repeat
LogoFileName = FileLocation ^ "logo.gif")
```

**Note:** To show the status of the request, instead of this repeat/while loop, you must use a series of icons to form the repeat loop: a calc icon to get the status, a display icon to show it, followed by a calc icon that loops, until `snxNetDone(NetID) = True`. See the example for `snxGetStreamStatus`.

See also `snxNetAbort`, `snxNetDone`, `snxNetError`, `snxGetStreamStatus`





## snxGetNetText

Syntax	Syntax1: <code>snxGetNetText (URL)</code> Syntax2: <code>snxGetNetText (URL, propertyList)</code>
Parameters	<i>URL</i> is the URL to retrieve. <i>PropertyList</i> is property list of parameters for CGI scripts.
Return	Success: unique id for the operation (> 0) Failure: error code (<0)
Description	<p>Initiates the loading of text from a file on an HTTP or HTTPS server, or initiates a CGI query.</p> <p>The URL must be properly formatted and must start with "http://" or "https://".</p> <p>If you use syntax 2, parameters in <i>propertyList</i> don't have to be URL encoded.</p> <p>To access a file on a HTTPS server, depending on how the server is configured, you may need to specify a user name and password within the URL, using the standard URL syntax.</p> <p>The length of the request (number of characters for the URL, plus parameters) is limited to the maximum allowed by Internet Explorer.</p> <p>Use <code>snxNetDone</code> to find out when the <code>snxGetNetText</code> operation is complete. Use <code>snxNetError</code> (or the Error Manager script) to find out if the operation was successful. Use <code>snxNetTextResult</code> to return the text downloaded by <code>snxGetNetText</code>. Use <code>snxGetStreamStatus</code> to get the bytes downloaded so far.</p>
Examples	<pre>-- accessing HTTPS with username and password -- required for destination server (NOT the proxy server) NetID := snxGetNetText ("https://eric:INM@www. integrati onnewmedi a.com") repeat while (snxNetDone(NetID) = False) end repeat ResultText := snxNetTextResult(NetID)  -- Syntax 1: passing parameters to an ASP script NetID := snxGetNetText("http://www. integrati onnewmedi a.com/query.asp?name=eri c%20Cartman") -- Syntax 2: passing parameters to an ASP script NetID := snxGetNetText("http://www.i ntegrati on. qc. ca/query.asp", [#name: "eric cartman"])</pre>
See also	<code>snxNetAbort</code> , <code>snxNetDone</code> , <code>snxNetError</code> , <code>snxNetTextResult</code> , <code>snxGetStreamStatus</code>

**Note:** To show the status of the request, instead of this repeat/while loop, you must use a series of icons to form the repeat loop: a calc icon to get the status, a display icon to show it, followed by a calc icon that loops, until `snxNetDone(NetID) = True`. See the example for `snxGetStreamStatus`.



## snxGetStreamStatus

**Syntax** `snxGetStreamStatus (NetID)`

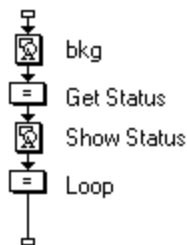
**Parameters** *NetID* is the unique network operation ID returned by the `snxGetNetText` or `snxDownloadNetThing` function.

**Return** Success: propertyList that contains the status of the network operation ([#URL:"xxx", #state: "yyyy", #bytesSoFar: z, #bytesTotal: u, #error: e])  
*#URL*: the current URL  
*#state*: current state. The value can be Connecting, Started, InProgress, Complete or Error.  
*#bytesSoFar*: number of bytes transferred.  
*#bytesTotal*: total number of bytes to transfer. This value may be equal to *#bytesSoFar* if the server doesn't include the content length in the header.  
*#Error*: Empty string if the download is not complete, "OK" if it completed successfully, or an error code (<0) if the operation ended with an error.  
 Failure: error code (<0)

**Description** Reads the status of the network operation identified by *NetID*.

**Examples** -- initiate a request to download a file from a URL  
`NetID := snxDownloadNetThing(myURL, FileLocation^"myDoc. PDF")`

icons used to show the status of a network operation, initiated previously



```

-----
-- in calc icon, Get Status, put the following script to Get the
status
if snxNetDone(NetID) <> FALSE then
  theError := snxNetError(NetID)
  --If an error occurs, display the error and retry connection
  if theError <> "OK" then
    GoTo(IconID@"Error Manager")
  end if
  GoTo (IconID@"Done Downloading") -- break out of repeat loop
else
  -- get the download status
  IStatus := snxGetStreamStatus (NetID) -- complete status list
  bytesSoFar := ValueAtIndex(IStatus, 3)
  bytesTotal := ValueAtIndex(IStatus, 4)
  theError := ValueAtIndex(IStatus, 5)
  if theError <> "OK" then
    GoTo(IconID@"Error Manager")
  end if
end if
-----
-- in display icon, Show Status, display the relevant variables
The complete property list would look something like this:
[#URL: "http://www.integrationnewmedia.com/documents/test.pdf",
 #state: "InProgress", #bytesSoFar: 3501, #bytesTotal: 50000,
 #error: ""]
-- in calc icon, Loop:
GoTo(IconID@"Get Status")
  
```



See also `snxNetDone`, `snxNetTextResult`, `snxGetNetText`, `snxDownloadNetThing`

## snxNetAbort

**Syntax** `snxNetAbort (NetID)`

**Parameters** *NetID* is the unique network operation ID returned by `snxGetNetText`, `snxPostNetText` or `snxDownloadNetThing` function.

**Return** Nothing.

**Description** Cancel a network operation (initiated with `snxGetNetText`, `snxPostNetText` or `snxDownloadNetThing`) identified by *NetID*. It has no effect if the operation was already complete.

**Examples** `snxNetAbort(NetID)`  
`GoTo(IconID@"Abort")`

See also `snxGetNetText`, `snxPostNetText`, `snxDownloadNetThing`

## snxNetDone

**Syntax** `snxNetDone (NetID)`

**Parameters** *NetID* is the unique network operation ID returned by `snxGetNetText`, `snxPostNetText` or `snxDownloadNetThing` functions.

**Return** TRUE if the network operation is completed with success or with an error; FALSE if the network operation is still in progress.

**Description** Indicates if a network operation (initiated with `snxGetNetText`, `snxPostNetText` or `snxDownloadNetThing`) identified by *NetID* is finished.

**Examples** `if snxNetDone(NetID) = True then`  
                  `ResultText := snxNetTextResult(NetID)`  
                  `GoTo(IconID@"DisplayResults")`  
`end if`

See also `snxNetTextResult`, `snxGetNetText`, `snxPostNetText`, `snxDownloadNetThing`

## snxNetError

**Syntax** `snxNetError (NetID)`

**Parameters** *NetID* is the unique network operation ID returned by `snxGetNetText`, `snxPostNetText` or `snxDownloadNetThing` function.

**Return** "OK" if the network operation completed with success.  
An error code (<0) if the network operation didn't complete with success.  
Empty string if the network operation is in progress or if *NetID* doesn't refer to a valid network operation.

**Description** Determines if an error occurred in a network operation



**Examples**

```

if snxNetDone(NetID) = True then
    if snxNetError(NetID) <> "OK" then
        theError := snxNetError(NetID)
        GOTO(IconID@"Error Manager")
    else
        ResultText := snxNetTextResult(NetID)
        GoTo(IconID@"DisplayResults")
    end if
end if

```

**See also** `snxNetDone`, `snxNetTextResult`, `snxGetNetText`, `snxPostNetText`, `snxDownloadNetThing`

## snxNetTextResult

**Syntax** `snxNetTextResult (NetID)`

**Parameters** *NetID* is the unique network operation ID returned by `snxGetNetText` or `snxPostNetText` function.

**Return** If successful: The text retrieved from the URL  
Otherwise: Empty string

**Description** Retrieves the text read during the network operation identified by *NetID*. After this function is called, the network ID no longer exists.

**Examples**

```

if snxNetDone(NetID) = True then
    if snxNetError(NetID) <> "OK" then
        theError := snxNetError(NetID)
        GOTO(IconID@"Error Manager")
    else
        ResultText := snxNetTextResult(NetID)
        GoTo(IconID@"DisplayResults")
    end if
end if

```

**See also** `snxNetDone`, `snxNetError`, `snxGetNetText`, `snxPostNetText`

## snxPostNetText

**Syntax** Syntax1: `snxPostNetText (URL, postText)`  
Syntax2: `snxPostNetText (URL, propertyList)`

**Parameters** *URL* is the URL to retrieve.  
*PostText* is the text to post.  
*PropertyList* is property list of parameters to post.

**Return** Success: unique id for the operation (> 0)  
Failure: error code (<0)

**Description** Initiates the posting of parameters to a CGI script on an HTTP or HTTPS server. The information is sent in the same way a browser posts an HTML form (using METHOD=POST).

The URL must be properly formatted and must start with "http://" or "https://".



To access a file on an HTTPS server, you must specify a user name and password using the standard URL syntax.

When using syntax 1 (passing parameters in a string), the parameters are sent to the server using MIME type "text/plain" and are not URL-encoded.

When using syntax 2 (passing parameters in a property list), the parameters are sent to the server using MIME type "application/x-www-form-urlencoded" and are URL-encoded.

Unlike with `snxGetNetText()`, there is no limitation on the number of characters in the request.

Use `snxNetDone` to find out when the `snxPostNetText` operation is complete. Use `snxNetError` (or the `ErrorManager` script) to find out if the operation was successful. Use `snxNetTextResult` to return the text returned by `snxPostNetText`. Use `snxGetStreamStatus` to get the bytes downloaded so far.

#### Examples

**Note:** To show the status of the request, instead of this repeat/while loop, you must use a series of icons to form the repeat loop: a calc icon to get the status, a display icon to show it, followed by a calc icon that loops, until `snxNetDone(NetID) = True`. See the example for `snxGetStreamStatus`.

```
-- accessing HTTPS where Username and password are required for
-- the destination server, NOT the proxy server
NetID := snxPostNetText ("https://Vahe:INM@www.
integrationnewmedia.com/orderform.cgi", [#color: "red",
#quantity: "200"])
repeat while (snxNetDone(NetID) = False)
end repeat
ResultText := snxNetTextResult(NetID)
GoTo(IconID@"DisplayResults")
```

See also `snxNetAbort`, `snxNetDone`, `snxNetError`, `snxNetTextResult`, `snxGetStreamStatus`

## snxProxyServer

**Syntax** Syntax1: `snxProxyServer ()`  
Syntax2: `snxProxyServer (Command)`  
Syntax3: `snxProxyServer (IPAddress[, Port[, UserName[, Password]]])`  
Syntax4: `snxProxyServer (#Default, [UserName[, Password]])`

**Parameters** Syntax2: *command* is a symbol that indicates the action to be performed by `snxProxyServer`:

#Stop: stop using a proxy server for network operations

#Address: returns the address of the proxy server

#Port: returns the port used by the proxy

#UserName: returns the user name used by the proxy

#Password: returns the password used by the proxy

Syntax3:

*IPAddress* is the IP address or URL of the proxy server (String)

*Port* is the port number used by the proxy server (Integer). This value can be omitted (the default value is 80)



*UserName* is the user name to use with a proxy server that requires authentication. This value can be omitted (the default value is "")

*Password* is the password to use with a proxy server that requires authentication. This value can be omitted (the default value is "")

**Syntax4:**

#Default: Sets the proxy settings to Microsoft Internet Explorer settings.

*UserName* is the user name to use with a proxy server that requires authentication. This value can be omitted (the default value is "")

*Password* is the password to use with a proxy server that requires authentication. This value can be omitted (the default value is "")

**Return** Depends on the syntax used (see Parameters and Description).

**Description** Syntax1: Reads the proxy server settings into a propertyList ([#Address: "xxx", #Port: yyy, #UserName: "zzz", #Password: "uuu"])

If there is no proxy server, the property list returned is empty.

Syntax2: Executes an action (see Parameters for details)

Syntax3: Sets the proxy server configuration

Syntax4: Sets the proxy server configuration to use the system's default proxy server settings (IE settings on Windows).

**Examples**

```
--syntax 3 to set the proxy server
-- with specific IP address and port number
snxProxyServer("204.19.171.45", 8080, "Vahe", "INM")

--syntax 3 with URL of the proxy server
snxProxyServer("http://myDomain.com/myProxy", 8080, "Vahe", "INM")

-- syntax 4 to set the proxy server with IE settings
snxProxyServer(#Default, "Vahe", "INM")

-- syntax 1 to read the proxy server settings into a property list
lProxySettings := snxProxyServer()
-- ([#Address: "204.19.171.45", #Port: 8080, #UserName: "Vahe",
#Password: "INM"]

-- syntax 2 to read the proxy server IP address
ProxyAddress := snxProxyServer(#Address)
-- "204.19.171.45"
```



## Appendix 2: Error codes

---

- 400: HTTP error: Invalid syntax.  
The request could not be processed by the server due to invalid syntax.
- 401: HTTP error: Access denied.  
The requested resource requires user authentication. User name and/or password may be absent or incorrect in the URL.
- 403: HTTP error: Request forbidden.  
The server understood the request, but is refusing to fulfill it.
- 404 HTTP error: Object not found.  
The server has not found anything matching the requested URI (Uniform Resource Identifier).
- 405 HTTP error: Method is not allowed.  
The method used is not allowed.
- 406 HTTP error: No response acceptable to client found.  
The server didn't find any response acceptable to the client.
- 407 HTTP error: Proxy authentication required.  
You must provide a valid user name and password to the proxy server using the snxProxyServer method.
- 408 Server timed out waiting for request.  
HTTP error: The server didn't receive the request. Try again.
- 409 HTTP error: User should resubmit with more information.  
The request could not be completed due to a conflict with the current state of the resource. You should resubmit with more information.
- 410 HTTP error: The resource is no longer available.  
The requested resource is no longer available on the server, and no forwarding address is known.
- 411 HTTP error: Content length required.  
The server refuses to accept the request without a defined content length.
- 412 HTTP error: Failed to evaluate precondition in the request.  
The precondition given in one or more of the request header fields evaluated to false when it was tested on the server.
- 413 HTTP error: Request entity too large.  
The server is refusing to process a request because the request entity is larger than the server is willing or able to process.
- 414 HTTP error: Request URI too long.



- The server is refusing to service the request because the request URI (Uniform Resource Identifier) is longer than the server is willing to interpret.
- 415 HTTP error: Unsupported media type.  
The server refuses to service the request because the request is in a format not supported by the requested resource for the requested method.
- 500 HTTP error: Internal server error.  
The server encountered an unexpected condition that prevented it from fulfilling the request.
- 501 HTTP error: Request not supported.  
The server does not support the functionality required to fulfill the request.
- 502 Error response received from gateway.  
HTTP error: The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
- 503 HTTP error: Temporarily overloaded.  
The server is temporarily overloaded.
- 504 HTTP error: Timed out waiting for gateway.  
The request was timed out waiting for a gateway.
- 505 HTTP error: HTTP version not supported.  
The server does not support, or refuses to support, the HTTP protocol version that was used in the request message.
- 965 Socket library error: Resource temporarily unavailable.  
This error is reported by the Socket library. The operation cannot be completed immediately. It should be retried later.
- 1001 Internet library error: Out of handles.  
This error is reported by the Internet library. No more handles could be generated at this time.
- 1002 Internet library error: Timeout.  
This error is reported by the Internet library. The request has timed out.
- 1003 Internet library error: Extended error.  
This error is reported by the Internet library. An extended error was returned from the server. This is typically a string or buffer containing a verbose error message.
- 1004 Internet library error: Internal error.  
This error is reported by the Internet library. An internal error has occurred.
- 1005 Internet library error: Invalid URL.  
This error is reported by the Internet library. The URL is not properly formatted. Here are some examples: <http://www.integrationnewmedia.com>  
<https://user:password@www.integrationnewmedia.com>
- 1006 Internet library error: Unrecognized scheme.





- This error is reported by the Internet library. The URL scheme could not be recognized or is not supported.
- 1007 Internet library error: Server name not resolved.  
This error is reported by the Internet library. The server name could not be resolved.
- 1008 Internet library error: Protocol not found.  
This error is reported by the Internet library. The requested protocol could not be located.
- 1009 Internet library error: Invalid option.  
This error is reported by the Internet library. A request to InternetQueryOption or InternetSetOption specified an invalid option value.
- 1010 Internet library error: Bad option length.  
This error is reported by the Internet library. The length of an option supplied to InternetQueryOption or InternetSetOption is incorrect for the type of option specified.
- 1011 Internet library error: Option cannot be set.  
This error is reported by the Internet library. The request option cannot be set, only queried.
- 1012 Internet library error: Shutdown.  
This error is reported by the Internet library. The Internet function support is being shut down or unloaded.
- 1013 Internet library error: Incorrect user name.  
This error is reported by the Internet library. The request to connect and log on to a FTP server could not be completed because the supplied user name is incorrect.
- 1014 Internet library error: Incorrect password.  
This error is reported by the Internet library. The request to connect and log on to a FTP server could not be completed because the supplied password is incorrect.
- 1015 Internet library error: Login failure.  
This error is reported by the Internet library. The request to connect to and log on to a FTP server failed.
- 1016 Internet library error: Invalid operation.  
This error is reported by the Internet library. The requested operation is invalid.
- 1017 Internet library error: Operation canceled.  
This error is reported by the Internet library. The operation was canceled, usually because the handle on which the request was operating was closed before the operation completed.
- 1018 Internet library error: Incorrect handle type.  
This error is reported by the Internet library. The type of handle supplied is incorrect for this operation.
- 1019 Internet library error: Incorrect handle state.  
This error is reported by the Internet library. The requested operation cannot be carried out because the handle supplied is not in the correct state.



- 1020 Internet library error: Not a proxy request.  
This error is reported by the Internet library. The request cannot be made via a proxy.
- 1021 Internet library error: Registry value not found.  
This error is reported by the Internet library. A required registry value could not be located.
- 1022 Internet library error:  
Windows: Bad registry value.  
Mac: Bad Configuration in Control Panels > TCP/IP.  
This error is reported by the Internet library.  
Windows: A required registry value was located but is an incorrect type or has an invalid value.  
Mac: Make sure your TCP/IP settings are correct or see your system administrator.
- 1023 Internet library error: No direct access.  
This error is reported by the Internet library. Direct network access cannot be made at this time.
- 1024 Internet library error: No context.  
This error is reported by the Internet library. An asynchronous request could not be made because a zero context value was supplied.
- 1025 Internet library error: No callback function.  
This error is reported by the Internet library. An asynchronous request could not be made because a callback function has not been set.
- 1026 Internet library error: Request pending.  
This error is reported by the Internet library. The required operation could not be completed because one or more requests are pending.
- 1027 Internet library error: Incorrect format.  
This error is reported by the Internet library. The format of the request is invalid.
- 1028 Internet library error: Item not found.  
This error is reported by the Internet library. The requested item could not be located.
- 1029 Internet library error: Cannot connect.  
This error is reported by the Internet library. The attempt to connect to the server failed. Could be a wrong port number, or server is busy.
- 1030 Internet library error: Connection aborted.  
This error is reported by the Internet library. The connection with the server has been terminated. Could be host is down or network is down.
- 1031 Internet library error: Connection reset.  
This error is reported by the Internet library. The connection with the server has been reset.
- 1032 Internet library error: Force retry.  
This error is reported by the Internet library. Calls for the Internet function to redo the



request.

- 1033 Internet library error: Invalid proxy request.  
This error is reported by the Internet library. The request to the proxy was invalid.
- 1036 Internet library error: Handle already exists.  
This error is reported by the Internet library. The request failed because the handle already exists.
- 1037 Internet library error: Security certificate date expired.  
This error is reported by the Internet library. SSL certificate date that was received from the server is bad. The certificate is expired.
- 1038 Internet library error: Security certificate common name invalid.  
This error is reported by the Internet library. SSL certificate common name (host name field) is incorrect. For example, if you entered `www.server.com` and the common name on the certificate says `www.different.com`.
- 1039 Internet library error: HTTP to HTTPS on redirect.  
This error is reported by the Internet library. The application is moving from a non-SSL to a SSL connection because of a redirect.
- 1040 Internet library error: HTTPS to HTTP on redirect.  
This error is reported by the Internet library. The application is moving from a SSL to a non-SSL connection because of a redirect.
- 1041 Internet library error: Mixed security.  
This error is reported by the Internet library. Indicates that the content is not entirely secure. Some of the content being viewed may have come from unsecured servers.
- 1042 Internet library error: Change post is non secure.  
This error is reported by the Internet library. The application is posting and attempting to change multiple lines of text on a server that is not secure.
- 1043 Internet library error: Post is non-secure.  
This error is reported by the Internet library. The application is posting data to a server that is not secure.
- 1044 Internet library error: The certificate on the server has expired.  
This error is reported by the Internet library. URL cannot be accessed without a valid certificate.
- 1045 Internet library error: SSL Protocol not supported by URL Access.  
This error is reported by the Internet library. If you see this error, you may be able to fix it by trying your request again. If it doesn't work, try adding a certificate to your server that's supported by URL Access. If that doesn't help, there is currently no other way to workaround this error. You should not see this error when running on Mac OS X 10.1 and later. (Mac only)
- 1046 Internet library error: The server requires 128-bit encryption, but URL Access doesn't support it.



- This error is reported by the Internet library. URL Access began supporting 128-bit encryption in version 2.1, which shipped with Mac OS 9.1
- 1047 Internet library error: File empty error.  
This error is reported by the Internet library. Indicates a failed data transfer operation.
- 1150 Internet library HTTP Error: Header not found.  
This error is reported by the Internet library. The requested header could not be located.
- 1151 Internet library HTTP Error: Downlevel server.  
This error is reported by the Internet library. The server did not return any headers.
- 1152 Internet library HTTP Error: Invalid server response.  
This error is reported by the Internet library. The server response could not be parsed.
- 1153 Internet library HTTP Error: Invalid header  
This error is reported by the Internet library. The supplied header is invalid.
- 1154 Internet library HTTP Error: Invalid query request.  
This error is reported by the Internet library. The request made to HttpQueryInfo is invalid.
- 1155 Internet library HTTP Error: Header already exists.  
This error is reported by the Internet library. The header could not be added because it already exists.
- 1156 Internet library HTTP Error: Redirect failed.  
This error is reported by the Internet library. The redirection failed because either the scheme changed (for example, HTTP to FTP) or all attempts made to redirect failed (default is five attempts).
- 1201 Internet library error.  
On Windows: Failed to load Internet Explorer library (wininet.dll).  
On Mac: URL Access failed to load or the URL Access library is unavailable.  
Windows: Make sure Internet Explorer 4.x or greater is installed on the computer and that the computer has an active Internet connection.  
Mac: Check that URL Access is in the Control Panels > Extensions Manager.
- 1202 Internet Explorer library error: Failed to set the callback procedure.  
Severe error that can be caused by an old or invalid version of wininet.dll.
- 1203 Internet library error: Failed to authenticate the user.  
The user name and/or the password defined with the snxProxyServer are not accepted by the proxy server.
- 1204 Internet library error: Failed to close the request.  
Failure attempting to close the request handle. System error. Close the application and restart to release this handle.
- 1205 Internet Config. error: failed to start or to find the config file.  
Check if there is "Internet" in the Control Panels and if you can set properties.



- 1352      Server Exception: Routing exception.  
Server is unable to forward the request to the appropriate target. See your system administrator.
- 1353      Server Exception: Relationship exception.  
The trust relationship between this workstation and the primary domain failed. See your system administrator.
- 1354      Server Exception: Gateway exception.  
Default Gateway is not properly configured. See your system administrator.
- 1355      Server Exception: IP Address exception.  
IP Address is not properly configured. See your system administrator.
- 1356      Server Exception: Server exception.  
Server is not properly configured. See your system administrator.
- 1357      Server Exception: Invalid status exception.  
Invalid status reported by server.
- 1358      Server Exception: Configuration exception.  
Server configuration can't be loaded.
- 1359      Server Exception: Socket exception.  
Cannot make client socket non-blocking.
- 2001      Internal error: Failed to allocate memory.  
The system is running out of memory. Try to close some applications or restart your computer.
- 2002      Internal error: Invalid port number.  
The proxy port number must be an integer greater than or equal to 0 and less than or equal to 65537.
- 2003      Internal error: Failed to generate a temporary file name.  
The system is not able to generate a unique temporary file name because too many temp files are in use. Close the application and restart.
- 2004      Internal error: Failed to create a temporary file.  
The working directory of the application may be on a read-only disk.
- 2005      Internal error: Failed to open the temp file.  
The temp file may have been deleted or moved manually from its location. Close the application and restart.
- 2006      Internal error: Failed to read/write to the temp file.  
The temp file may have been deleted or moved manually from its location. Close the application and restart.
- 2007      Internal error: Invalid network operation ID.



- The network operation ID doesn't exist. Make sure you are using a valid ID and that you have not disabled this ID (for example, by calling `snxNetTextResult`).
- 2008 Internal error: No URL specified.  
You must provide a valid URL.
- 2009 Internal error: Internal failure.  
Severe error; may be caused by a memory problem. Restart the computer.
- 2100 Internal error: Unsupported protocol.  
INM's SecureNet only supports HTTP and HTTPS protocols. Make sure that the URL starts with `http://` or `https://`.
- 2101 Normal Operation: Request aborted by user.  
User purposefully aborted the network operation.
- 2102 Syntax Error: Invalid file name.  
Local destination file name for downloaded file is not a valid file name.
- 2101 Wrong mode for this Xtra.  
This error is triggered when `snxDownloadNetThing` is used in Shockwave.
- 2102 Not supported on Mac OS 10.1.x.  
This error is generated when calling `snxProxyServer()` to initialize a proxy server on Mac OS 10.1.x.
- 2103 Not supported on Mac OS 10.1.x. when using a proxy.  
This error is generated when trying to perform a HTTP or secure HTTP operation across a proxy server on Mac OS 10.1.x.



# Index

---

## B

Basic tasks, 13

## D

Delivering to end user, 20

Demos, 6, 23

Discussion list, 23

Downloading files, 14

## E

Errors

    Debugging, 16

    Error Manager sample,  
    16

Example

    Posting data to a form, 15

    Requesting data using  
    Get, 14

## F

Functions

    snxDownloadNetThing,  
    24

    snxGetNetText, 14, 25

    snxGetStreamStatus, 15,  
    26

    snxNetAbort, 27

    snxNetDone, 15, 27

    snxNetError, 27

    snxNetTextResult, 28

    snxPostNetText, 15, 28

    snxProxyServer, 29

Functions reference  
(Appendix 1), 24

## I

Internet Explorer

    proxy server settings, 11

## L

Licensing, 9

## M

Mac

    Requirements, 8

    Web-Packaging, 20

Mailing list, 23

## P

Packaged pieces, 20

Posting data, 15

Proxy server

    definition, 10

Proxy server settings, 10

## R

Registration, 9

Requesting data, 14

Requirements

    Authorware, 8

    Internet Explorer, 8

    Windows, 8

## S

System Requirements, 8

## T

Technical support, 23

TechNotes, 23

Troubleshooting

    Code, 16

    Network connections, 18

Tutorials, 6, 23

## U

URLs

    Encoding, 13

    Formatting, 13

## W

Web-Packaging, 20

Windows, 8

